VOCCluster: Untargeted Metabolomics Feature Clustering Approach for Clinical Breath Gas Chromatography - Mass Spectrometry Data

TITLE PAGE

VOCCluster: Untargeted Metabolomics Feature Clustering Approach for Clinical Breath Gas Chromatography - Mass Spectrometry Data

Yaser Alkhalifah[1], Iain Phillips[1], Andrea Soltoggio[1], Kareen Darnley[2], William H. Nailon[2], Duncan McLaren[2], Michael Eddleston[3], C. L. Paul Thomas[4] and Dahlia Salman[4*]

[1]Department of Computer science, Loughborough University, Loughborough, LE11 3TU, UK

[2]Edinburgh Cancer Centre, NHS Lothian, Edinburgh, UK

[3]Pharmacology, Toxicology & Therapeutics Unit, University of Edinburgh, Edinburgh, UK

[4]Department of Chemistry, Loughborough University, Loughborough, LE11 3TU, UK

*Corresponding author: Dahlia Salman (D.Salman@lboro.ac.uk)

Yaser Alkhalifah
Department of Computer science
Loughborough University
Loughborough UK LE11 3TU

Iain Phillips
Department of Computer science
Loughborough University
Loughborough UK LE11 3TU

Andrea Soltoggio
Department of Computer science
Loughborough University
Loughborough UK LE11 3TU

Kareen Darnley
Pharmacology, Toxicology & Therapeutics Unit
University of Edinburgh
Edinburgh, UK EH8 9YL

William H. Nailon
Pharmacology, Toxicology & Therapeutics Unit
University of Edinburgh
Edinburgh, UK EH8 9YL

Duncan McLaren
Pharmacology, Toxicology & Therapeutics Unit
University of Edinburgh
Edinburgh, UK EH8 9YL

Michael Eddleston
Pharmacology, Toxicology & Therapeutics Unit
University of Edinburgh
Edinburgh, UK EH8 9YL

C. L. Paul Thomas
Department of Chemistry
Loughborough University
Loughborough, UK LE11 3TU

Dahlia Salman **(corresponding author)**
Department of Chemistry
Loughborough University
Loughborough, UK LE11 3TU
D.Salman@lboro.ac.uk

VOCCluster: Untargeted Metabolomics Feature Clustering Approach for Clinical Breath Gas Chromatography - Mass Spectrometry Data

Yaser Alkhalifah[1], Iain Phillips[1], Andrea Soltoggio[1], Kareen Darnley[2], William H. Nailon[2], Duncan McLaren[2], Michael Eddleston[3], C. L. Paul Thomas[4] and Dahlia Salman[4]*

[1]Department of Computer science, Loughborough University, Loughborough, LE11 3TU, UK

[2]Edinburgh Cancer Centre, NHS Lothian, Edinburgh, UK

[3]Pharmacology, Toxicology & Therapeutics Unit, University of Edinburgh, Edinburgh, UK

[4]Department of Chemistry, Loughborough University, Loughborough, LE11 3TU, UK

*Corresponding author: Dahlia Salman (D.Salman@lboro.ac.uk)

ABSTRACT: Metabolic profiling of breath analysis involves processing, alignment, scaling and clustering of thousands of features extracted from Gas Chromatography Mass spectrometry (GC-MS) data from hundreds of participants. The multi-step data processing is complicated, operator error-prone and time-consuming. Automated algorithmic clustering methods that are able to cluster features in a fast and reliable way are necessary. These accelerate metabolic profiling and discovery platforms for next generation medical diagnostic tools. Our unsupervised clustering technique, VOCCluster, prototyped in Python, handles features of deconvolved GC-MS breath data. VOCCluster was created from a heuristic ontology based on the observation of experts undertaking data processing with a suite of software packages. VOCCluster identifies and clusters groups of volatile organic compounds (VOCs) from deconvolved GC-MS breath with similar mass spectra and retention index profiles. VOCCluster was used to cluster more than 15,000 features extracted from 74 GC-MS clinical breath samples obtained from participants with cancer before and after a radiation therapy. Results were evaluated against a panel of ground truth compounds and compared to other clustering methods (DBSCAN and OPTICS) that were used in previous metabolomics studies. VOCCluster was able to cluster those features into 1081 groups (including endogenous, exogenous compounds and instrumental artefacts) with an accuracy rate of 96% (± 0.04 at 95% confidence interval).

Breathomics, the analysis of volatile organic compounds (VOCs) in breath, offers a promising approach for the non-invasive study of metabolic processes and derangements[1]. Much has been made of its potential for the development of new and enhanced diagnostic approaches[2]. Non-targeted metabolomic studies with breathomics use Gas Chromatography-Mass Spectrometry (GC- MS) as the gold standard analytical technique[3]. The combination of high-resolution separations, low limits of detection (picogram level) and mass spectral fragmentation patterns provides efficient Class 2 compound identification[4] (Putatively annotated compounds that were identified based on mass spectral similarities without chemical reference standards). Amann and Smith (2013) provide an excellent introduction to the theory and practice of GC-MS in breathomics[2]. Despite the high fidelity of GC-MS breathomics data, it is not yet possible to adopt and follow the guidelines and recommendations for metabolomic characterisation proposed and adopted widely in metabolomic studies involving blood-plasma or urine[5]. The reasons for this "arrested- development" arise from the nature of breath samples and the inherent variability of GC-MS data.

Breath samples are not stable and consequently cannot be stored for significant lengths of time[6]. This means that pooled samples and batch processing are not currently possible. Hence, breath data are acquired throughout the study and contain the artefacts that arise from instrument degradation and maintenance cycles. A putative workflow has described how these attributes of breath samples may be managed[7]. This previous work also described how deconvolution of the mass spectra obtained from co-eluting VOCs and their subsequent registration through retention indexing could be used to assign unique identifiers to unknown VOCs and thereby facilitate multi-variate analysis. However this work was incomplete as it did not: address adequately the variability of the mass spectrometric component in GC-MS breath data; and, it did not solve the impracticality of scaling the workflow to encompass the many tens of thousands of breath features that are generated from even a modestly-sized study (cohort size: n=20 to 50).

Deconvolution generates a retention-indexed (RI) mass spectra and peak area for each isolated feature (supplementary data, Figure S1). An ideal analysis would result in any given VOC generating identical mass spectra each time and consequently being assigned the same identifier. However small variations in the intensities of the mass-to-charge ratios (m/z) of the spectrum's fragment ions results in the potentially for the same VOC feature being assigned an alternative identity. Consequently, post-processing of breath data currently requires specialist, expert evaluation of every extracted feature to verify that each VOC is correctly classified giving a single identifier across the breath data matrix as a prelude to multi-variate modelling. This inherently human-based methodology is not sustainable when studying untargeted metabolomics involving the thousands of VOCs in the breath samples in the study. The endeavour turns rapidly into an exercise in human endurance with the further introduction of variability and misclassification errors arising from operator fatigue or knowledge gaps.

This research was fostered by previous classification approaches that proposed supervised pattern recognition methods for the classification of breathomics GC-MS data. Van Berkel *et al*[8] applied a Support Vector Machine (SVM) algorithm to breath data to distinguish between smoking and non-smoking subjects (n = 22); by creating predictive models with significant generalisation power despite working with small data set or data with large variation. However, a SVM is only feasible when classification labels are available, and this is not the case when working with blind clinical trials where labels are revealed only once the potential biological candidates are discovered. Generally, such supervised techniques fail to show the results of VOCs clustering from different samples. Therefore, separability concepts and unsupervised clustering are implemented with the aim of identifying the smallest subset of exhaled metabolites which could deliver the most robust and precise predictions with regards to the clinical phenotype of interest.

Prior research shows that some of the common statistical tactics included primary separation methods such as the statistical hypothesis test (t-test), which was used in the identification of important VOCs[9]. Principal component analysis (PCA) was used in previous metabolomic studies as it improved the human perception of data through the reduction of space for multidimensional data[10-11], and correlation analysis was employed in the detection of respiratory infections and others[12-13]. More sophisticated unsupervised techniques for data mining were employed previously for the purposes of analysing multivariate data through the production of clusters. Examples include K-means[14], Ordering Points To Identify the Clustering Structure (OPTICS)[15], Density-Based Spatial Clustering of Applications with Noise (DBSCAN)[16], and hierarchical clustering (HC)[17]. Unsupervised techniques are based on a distance or similarity measurement. Manhattan, Euclidean, and Cosine are examples of multivariate distances or similarity measurements[17].

Available algorithms are either designed for *targeted* analysis, where only a panel of compounds is searched for and clustered together, or *untargeted* analysis which is user dependent and the results (i.e. number and size of clusters) are influenced by given parameters. Untargeted analysis causes additional difficulties as the parameters are often estimated and decided by the user and could be unconsciously influenced by the needed results[19]. DBSCAN, OPTICS and HC methods are examples of the most commonly used clustering methods in untargeted metabolomics data analysis. All these algorithms do not require *k*, an estimate of the number of the clusters, in advance and are therefore useful in untargeted studies where the number and size of clusters are unknown. Susceptibility to outliers with the number of clusters and the accuracy of each cluster is however the challenge, when using these clustering techniques with high-dimensional breathomics data and large sample size. For example, K-means clustering technique will struggle with breath data as it assumes that the distribution of objects in each cluster is spherically distributed around the centre[20]. This could result in poor clustering performance when used for data with outliers, or with shapes that are not spherical or clusters that have different sizes[19], such as is the case with breathomics data. HC is another example which differs from K-means clustering as it does not deliver a single dataset partition. HC is based on the consolidation of peak lists by calculating the distances of metabolites to produce groups of same multivariate similarity[21,22]. De Souza *et al* (2006) used HC for GC-MS data collected for the metabolic profiling of Leishmania parasites[21]. However, HC requires the scientist to make an arbitrary decision of how and where to cut the presented dendrogram, which becomes harder with hundreds of thousands of VOCs detected per cohort.

DBSCAN was reported by Ester *et al.* (1996) as a density-based clustering process in a time determined similarity matrix[22]. It is used for clustering tandem mass spectra data for both metabolomics and proteomics fields[24-25]. However, challenges with DBSCAN were reported when the data are highly dimensional with varying density profiles between clusters[26]. OPTICS clustering was used by Depke *et al* (2017) to develop CluMSID algorithm which was applied to cluster mass spectra of *P*. aeruginosa cell extract[27]. The CluMSID workflow was reported to provide correctly grouped metabolites with common functional elements such as peptides. However, this algorithm was not developed or used for highly dimensional variable breath data.

Despite both supervised classification and unsupervised clustering techniques described above, the algorithms are specific to the nature of data and research question. These approaches may not necessary enable mechanistic molecular identity or a specific metabolic pathway recognition. Therefore, there is a need for an algorithm that is automated, fast, not user dependent and can cluster thousands of similar VOCs from hundreds of samples while taking both RI and m/z variation between samples into considerations.

The current study sought to capture the heuristic ontology used by breathomics researchers in clustering deconvolved breathomics features, so that features that arise from identical chemical species are assigned to a unique identifier and clustered together correctly; regardless of variability in signal attributes. Their knowledge was used to iteratively develop a computational clustering method that produces the essential feature clustering step for GC-MS breathomics data that resolves the scalability roadblock and enhances the reliability of the clustering encoding. This is the vital step in the preparation of the "discovery" data modelling. OPTICS and DBSCAN were initially used to cluster breath data. The results of this experiment, which is also presented in this manuscript, have guided us to implement a novel unsupervised feature clustering technique, VOCCluster.

The proof of concept of the algorithm VOCCluster is coded in Python. VOCCluster identifies clusters of the same VOCs from different deconvolved breath samples in a non-supervised manner. It employs cosine similarity between VOCs' mass spectra within a retention index (RI) region from different samples to establish a distance measurement. Cosine similarity has been used as a robust similarity measure when compared with other measuring distances such as Euclidian distance, centroid etc [28-30]. As opposed to other clustering techniques, VOCCluster continuously monitors the clustering of features and reassesses cluster membership as the algorithm progresses. VOCs can be removed and re-clustered, which involves additional computational processes, but improves the potential quality of the candidate clusters.

In this paper, VOCCluster was tested to process clinical breath samples obtained from participants with cancer before and after a radiation therapy dose. This research formed part of the TOXI-triage project's clinical trial[31]. VOCCluster's performance was compared with DBSCAN and OPTICS. Each was tuned, modified and applied to the same data and the results of all three algorithms were compared and evaluated. A panel of compounds that were extracted and clustered manually by an expert were used to evaluate the accuracy of these techniques.

MATERIALS AND METHODS

VOCCluster measures mass spectra similarities for VOCs from different samples and forms a similarity matrix. This is supplemented by an examination of the RI for each VOC and together these are used to cluster VOCs into groups, each group containing a single VOC from each sample. Figure 1 illustrates and simplifies workflow stages of clustering breathomics deconvoluted data using VOCCluster.

Retention Index Variance Calculation (RIVC)

350-500 peaks are usually extracted from each breath sample including both endogenous and variable exogenous compounds. Retention time (RT) for each of these extracted peaks is not fixed as it shifts over the period of analysis, so it is possible to have different RTs for the same compound in different samples[32]. Therefore, there is a natural and expected variation in conditions between samples and there is a need to calibrate the expected range for retention index (RI) for each VOC in a

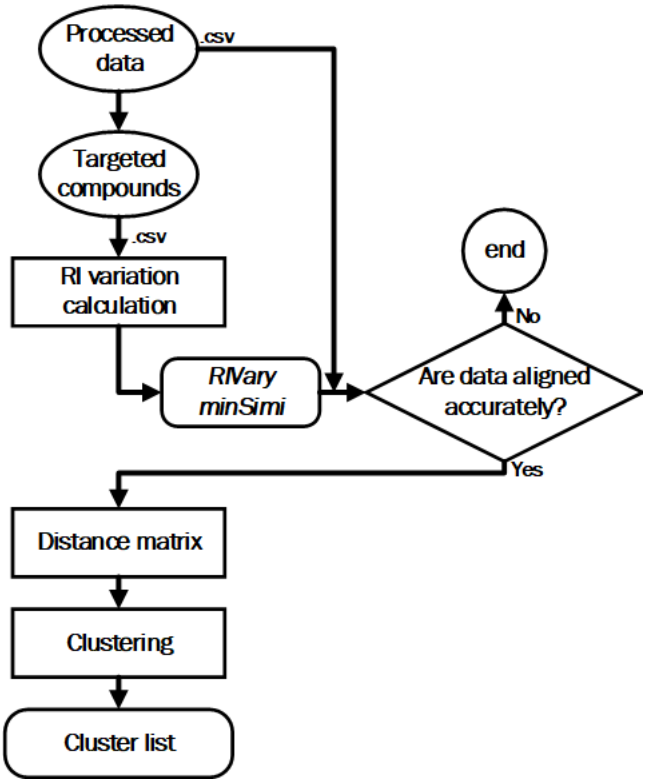


Figure 1. A workflow summarising the different processing steps to cluster the deconvoluted data. The first step starts by calculating the RI variations for a given input and thus the cosine similarity will be calculated between all the features in the given dataset. This process generates a report that operator checks the RIs variations between samples, correct any error and repeat the process.

Variation in RI for each VOC was still observed for the batch of samples that were calibrated together because of water retention in the samples and this in return affects the accuracy of clustering. This challenge is rather specific to breathomics samples and data processing and it is not observed in data from other matrices such as urine, blood etc. In this work, an estimation window of RI range for each VOC was proposed to be used for when clustering was carried out. Calculating the RI variations for the targeted compounds helps to build a reliable similarity matrix and consequently more accurate clustering outputs. It includes a retention index variance calculation (RIVC) method which calculates a RI variation between breathomics samples using an input list of targeted compounds (*targetedVOCs*) covering the whole range of the chromatogram. It is important to note that the *targetedVOCs* were only used for RIVC and not for clustering. The *targetedVOCs* help to identify the possible RI range for a VOC in a sample, which will then be needed for that VOC to be clustered with similar VOCs from other samples. The *targetedVOCs* can be formatted by selecting one compound from any breath sample (for each targeted compound) in the given dataset. The *targetedVOCs* table (e.g. Table S1 in supplementary data) includes VOCs that exist in a majority of samples as well as their estimated RI range. For example, spiked internal standard compounds and common endogenous and exogenous compounds. The RIVC method calculates the RI-range for each VOC in the *targetedVOCs* list. This discovered range helps to predict the possible RI range for other VOCs in the study.

The first step in the RIVC involves comparing cosine distances among compounds in the restricted set of those chosen as targets. This first preliminary clustering builds groups of target compounds across samples. The RI is identified for each target compound in each of the samples. For each group of the same targeted compound, the maximum and minimum RIs were noted, RI_{min_i} and RI_{max_i}. The RI space is then segmented based on the midpoints between successive RI_{max_i} and $RI_{min_{i+1}}$ positions. Each segment is then given a ΔRI_i value calculated as the difference between RI_{max} and RI_{min}. The segment RI points and ΔRI values were then stored to be used later in the algorithm. Any VOC (a) that appears in a segment region of targeted compound i will have a RI range that is be calculated using equation 1 below.

$$RIrange = \begin{cases} RI_a - \Delta RI_i \,, \\ RI_a + \Delta RI_i \,, \end{cases} \qquad \text{(Equation 1)}$$

RIVC also calculates the distances among all VOCs in each group at this stage to find the minimum available similarity between two VOCs at the same group. This is needed to determine the minimum threshold epsilon (ε) similarity of two VOCs to be assigned as same VOCs, which will be detailed later in section 2.3. In other words, ε is estimated for VOCCluster (unlike other approaches) and used as an input parameter for the clustering process.

In addition, RIVC generates a report at this step which includes clusters of all the targeted compounds that need to be evaluated by a human operator to examine if the samples were aligned correctly. The similarity matrix is then built, and the clustering process is executed. Otherwise, any samples with RI alignment errors that are observed by the operator have to be reprocessed and aligned again for the RI variations to be recalculated.

Similarity matrix of mass spectra (DMMS)

Using the RIVC outputs, similarity matrix of mass spectra (DMMS) method is able to narrow the choice of potentially similar VOCs in other samples as it will only search for VOCs within the calculated RI range. The first step is to calculate the cosine similarity (Equation 2[29]) between a particular VOC in a sample with other VOCs in different samples within the RI range.

$$D(A,B) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\ \sqrt{\sum_{i=1}^{n} B_i^2}} \qquad \text{(Equation 2)}$$

where A_i and B_i are the m/z intensities of mass A and B respectively. The distances between assigned VOCs are stored in a matrix. A similarity value of zero was given when compared VOCs were outside the RI range, otherwise the cosine similarity was calculated. The size of the cosine matrix is m × m, where m is the number of VOCs in the collection. Table S2 and Figure S2 in supplementary data show an example of a similarity matrix. The same similarity matrix is used in all clustering techniques examined in this study.

Clustering approaches in breathomics

To introduce the notion of clustering techniques in breathomics studies, we first make the following observation: consider a set of breath samples (N) that each contain a set of VOCs (points) that are listed in a dataset for clustering purposes. In breathomics studies, N is known and therefore, the number of points per cluster (C_{max}) should not exceed N, for example acetone will be detected in every breath sample and therefore there should be no more than N acetone compounds identified.

Breathomics datasets may contain clusters of varying density and in some cases, there are no density drops between close

clusters or they may even overlap. For example, (heptane, 2,4-dimethyl-) (A) and (octane) (B) (figure 2) may appear in some breath samples within the calculated RI thresholds and have similar mass spectrum profiles except for 2-3 ions of higher m/z values. Consequently, the clusters of these two different groups are difficult to differentiate.

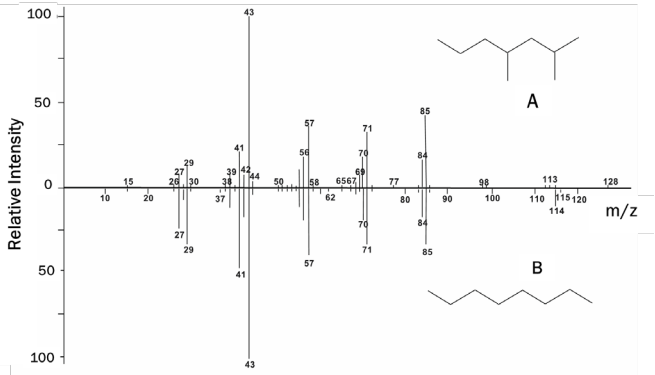


Figure 2. Mass spectra for (A) heptane, 2,4-dimethyl- and (B) octane, both compounds have similar mass spectrum profiles and appear within the calculated RI thresholds.

DBSCAN and OPTICS perform poorly in these circumstances. This fact is illustrated in figure 3, where A and B are density-based clusters with respect to C_{max}. Part of cluster A is classified with cluster B as a result of the high similarity between the points of the two clusters which should be in separate clusters. In other words, part of A points is in the neighbourhood of the B points. Therefore, these points have been clustered with B even though these points belong to A.

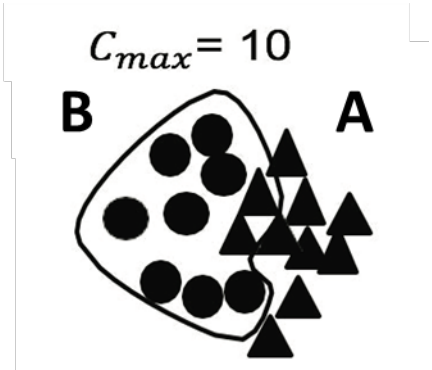


Figure 3. Illustration of the DBSCAN and OPTICS nested density-based clusters. B represents octane and A represents heptane, 2,4-dimethyl-. Part of A points have been clustered with cluster B.

As this scenario happens with DBSCAN and OPTICS, when points are in the neighbourhood of a point, they will be clustered with it even if these points might be closer to other points. Also, clustered points are never re-clustered again with these techniques, so the resultant clusters are strongly dependent on the ordering of the points into the clustering process. Example of such cases can also be found in Figure S3 in the supplementary data which shows the overlapped clusters and their shapes for some extracted compounds from the used dataset. Therefore, it is critical to determine the appropriate parameters as any deviations will cause different partitioning of the data set and consequently potential misclassification in the clusters.

VOCCluster algorithm

In VOCCluster several similarity properties are calculated at the time of processing each point. As the algorithm starts the clustering process, a candidate point is selected and then the points with the highest similarity values are collected and processed first. This will ensure that points grouped in a cluster all have the highest probability of similarity to that point and each other. Furthermore, a clustered point can be re-clustered into another cluster, if that point is highly similar to a point in the new cluster while still taking ε and other parameters into consideration. The novelty of VOCCluster (when compared to other techniques such as DBSCAN, OPTICS and HC) is the ability to monitor the clustering process and re cluster VOCs continuously as the algorithm progresses. Also, VOCCluster calculates ε (the similarity threshold) to use as an input parameter for the clustering process unlike other techniques where ε is chosen by the operator. VOCCluster therefore involves additional computational processes that improve the quality of the clustering outcome. Using octane and heptane, 2,4-dimethyl- as examples, Table S3 in supplementary materials shows VOCCluster re-clustering process in more details.

At the end of the clustering process, VOCCluster assigns cluster membership to each point in the dataset where outlier points will not be clustered. For the purpose of clustering, VOCCluster classifies points in the dataset as established, core, border and outlier points. The definition of each point is illustrated in figure 4 below.

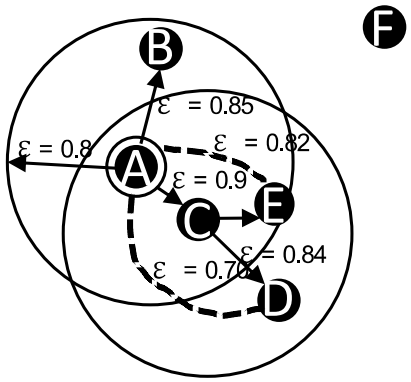


Figure 4. Illustration of the VOCCluster points definition. Point **A** represents an established point which is the first point that is processed in a cluster and has neighbours within similarity distance, **ε**. Point **C** is a core point because it is clustered and has neighbours within similarity distance **E** close to it more than to to the others. Points **B**, **E** and **D** are border points because they are not established or core points. Point **F** is an outlier point because it is not within similarity distance **ε** of any other point in the cluster.

VOCCluster stores information (properties) about each point before it is allocated to a cluster. These properties will be used to assess the similarities between the points in the dataset in order to cluster them. Moreover, some of these properties are also used to re-cluster any point that was clustered, when this point is closer to a point in another cluster than the core point that is at the same cluster. These properties consist of only five values for each clustered point: labelled, core-distance, core-point, established-distance and sampleId introduced in the following definitions:

- ***Labelled*** of a point is the cluster value that is given to this point where a subset of the same points that were clustered together from different samples are given the same value. Labelled property for the non-clustered object is undefined.

- ***core-distance*** of a point for a point p is the similarity value between p and the highest similarity point at the same cluster. For example, the *core-distance* value for point E in figure 4 is 0.9, which is the distance between E and the closest point in the cluster which is C as C was the highest similar point to E. *core-distance* for non-clustered points will be zero.
- ***core-point*** for a point p is the point with the highest similarity to p. For example, point C is the *core-point* for point E and D in figure 4. Un-clustered points do not have a core point.
- ***established-distance*** for a point p is the similarity value between p and the established point of a cluster. For example, the *establish-distance* for point C, E, D and B is A, where A is the first point that was processed in the cluster. Un-clustered points do not have an *establish- distance*.
- ***SampleId***, for a point p, is the biological sample that this object is belonging to. For example, if p is sourced from sample "1", then the *sampleId* for p is 1. The *sampleId* for each object has been provided in the dataset.

VOCCluster requires two parameters to be input, which are ε (which is calculated from 2.1) and minPts (which is the minimum neighbours to consider a point as an established point). VOCCluster selects a non-clustered point, p, from the dataset to start a new cluster (Figure S4). p will be assigned as an established point and p will be its own core-point. All of the neighbours' points of p, with respect to ε, are examined and stored in potentials list, ***N***, even if the points are already clustered and all of the points' properties are updated (Figure S5). The neighbour points in N are ordered based on the highest similarity value to p. If $|N| \geq minPts$ then starts a new cluster, Figure S6. Properties for all points that will be added to the cluster are updated, Figure S7. The most similar point in N list, q, will be examined to find its neighbours. N and properties of the points will be updated with the neighbours of q. Points that are already in N and have higher similarity to q than p will be updated. The process keeps repeating and the cluster grows until no points in N can be added or C_{max} is fulfilled. After that, a new un-clustered point will be selected from the main loop again to start a new cluster and the process repeated.

However, when a clustered point, w, is added into N, this indicates that w is an overlap point and VOCCluster will test the two properties values of the overlapped w. If w has a core-distance value to a point in the new cluster greater than core-distance value to a point in the previous cluster and, the establish-distance of w in the new cluster is greater than the establish-distance of w in the previous cluster, then, it will be moved to the new cluster and w's properties are updated accordingly. All of the points that share the same core-point, w, in the previous cluster will be re-clustered with w. Otherwise w will be left as it was. Applying this sophisticated movement of points between clusters improves the accuracy of clustering because a point will be allocated to the most probable similarity point in the dataset. The code used in these experiments is available at https://github.com/Yaser218/Untargeted-Metabolomics-Clustering.

The novelty of VOCCluster comes from the continuous process of re clustering of VOCs. These enable a VOC to be reassessed and cluster membership to be corrected if another VOC has a higher similarity to those in the cluster. This overcomes problems with existing techniques where once a VOC is clustered, it will never be re-clustered again. This makes VOCCluster less sensitive to the order of the VOCs in the dataset. Figure 3 demonstrated misclustering as part of the A cluster was clustered with the B cluster. VOCCluster is able to re-cluster these VOCs into the correct cluster.

Performance Evaluation

Experimental data sets

DBSCAN, OPTICS and VOCCluster were applied to cluster features for a set of clinical breath samples (n = 74) obtained from 24 patients with different types of cancer receiving radiation therapy dose as part of the TOXI-triage project[29]. Ethical approvals were sought and granted to collect breath samples from participants. The collected breath samples were analysed by thermal desorption - gas chromatography – quadrupole mass spectrometry (TD-GC-MS). Samples were analysed with thermal desorption (Unity-2, Markes International) interfaced to a GC (Agilent, 7890A) coupled to a quadrupole mass spectrometer (Agilent, MS 5977A), see Table S4 for operating details. The TD-GC-MS data were deconvolved to extract 350-500 VOC features per sample (AnalyzerPro Spectral Works, UK). Each breath sample was divided into four segments covering 2-5, 5-10, 10-20, and 20-45 min, and a deconvolution method was optimised per segment to minimise over or under-deconvolution. The features were aligned following Kovats retention indexing method (AnalyzerPro, Spectral Works, UK) and the retention index was calculated automatically for all the extracted features. The dataset contained 15,307 deconvoluted features. to be processed by VOCCluster. Additionally, a list of targeted compounds (supplementary data Table S1) is used in VOCCluster to calculate the RI variation. Both DBSCAN and OPTICS were modified to fulfil the conditions in section 2.3 and used for the same clinical breath data set. The clustering accuracy for a number of the ground truth compounds was evaluated. DBSCAN and OPTICS were tested using different thresholds (ε).

Sensitivity and specificity of clusters

The performance was evaluated by determining the accuracy of VOCs clusters according to the following:

- **True positive** (TP): a point clustered in the correct group.
- **False Positive** (FP): a point clustered in an incorrect group.
- **False Negative** (FN): a sample contains a point and it was not clustered
- **True Negative** (TN): a sample doesn't contain a point and this point wasn't clustered. This is limited to feature relating to points that exist in other samples.

A list of 27 *ground truth* compounds, covering a range of chemical functional groups and masses, was selected and used for comparison with the clusters generated by VOCCluster and the other algorithms. The experts were blinded to the results of the automated clustering when creating the list of the 27 ground truth compounds. The ground truth compounds were acetone; methane-d, trichloro; toluene; toluene-D8; cyclotetrasiloxane, octamethyl-; benzophenone; cyclopentasiloxane decamethyl-; cyclotrisiloxane, hexamethyl-; cyclohexasiloxane, dodecamethyl-; benzophenone 1,1':3',1"-terphenyl-2'-ol; benzaldehyde; benzene; nonanal; heptanal; decanal; ethylbenzene; à-pinene; hexanal; furfural; benzofuran; acetic acid; styrene; thiophene, 3-methyl-; 1-hexanol, 2-ethyl-; heptane, 2,4-dimethyl-; octane and 2,4-dimethyl-1-heptene. Some of these targeted compounds such as siloxanes were well isolated and easier to extract and cluster than other features that needed sophisticated deconvolution methods and careful considerations for the variation in mass spectra data. The accuracy of each cluster was calculated as shown in equation 3 below:

$$\frac{(TP + TN)}{(TP+TN+FP+FN)} \quad \text{(Equation 3)}$$

Furthermore, the mean of the distance similarity between VOCs in a cluster was calculated for each of the ground truth compounds and an overall similarity between the clustered VOCs was determined.

RESULTS AND EVALUATION

Accuracy of the manual retention indexing alignment step of the 74 samples was checked using the panel of targeted VOCs illustrated in figure 5. The RI-variation report was generated and entails the absolute clusters of those targeted VOCs. Correction of the RI alignment is important for the clustering phases, the smaller the RI-variation the faster and more accurate the clustering functions.

The calculated ΔRI (Figure 5) varied over the RI range. For examples ΔRI for acetone and cyclohexasiloxane, dodecamethyl- were 24 and 49 RI units, respectively. Therefore, the use of one or an estimated ΔRIs value for the entire analysis is not an accurate assumption and it is important to calculate the variation of the RIs per compound to help improve the accuracy of the clustering techniques functions. The mean, standard deviation and coefficient of variation (%) was calculated for RIs per each compound within the targeted VOCs panel, any sample with incorrect alignment was highlighted and corrected manually by re-deconvoluting/aligning the data using AnalyzerPro. Once alignment for that sample was corrected, the ΔRIs was recalculated and used to generate the distance matrix using the DMMS method. The generated distance matrix included many zeros, this is a result of either no obvious statistical correlation between the VOCs or their RIs were not compatible.

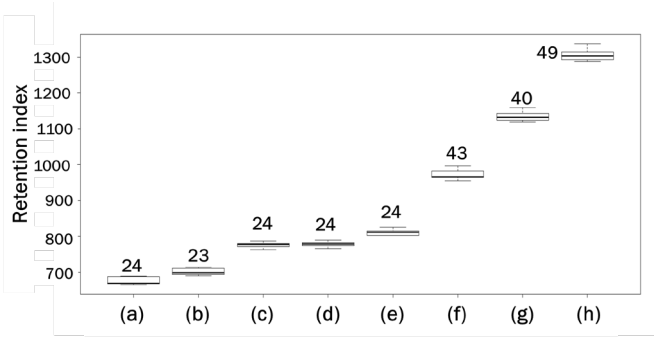


Figure 5: Box plots for the targeted compounds that were used to calculate the RI variations from multiple samples, dark line in each box represents the mean RI value for the compound. (a) acetone, (b) methane-d,trichloro-, (c) toluene-D8, (d) toluene, (e) cyclotetrasiloxane,hexamethyl-, (f) cyclotetrasiloxane,octamethyl-, (g) cyclotetrasiloxane,decamethyl-, (h) cyclotetrasiloxane,dodecamethyl-. Calculated ΔRIs were calculated per compound and is shown next to each box plot.

VOCCluster and the modified DBSCAN and OPTICS were used to cluster VOCs in the dataset. VOCCluster clustered all the deconvoluted features in approximately 3 hours. A similar number of samples is estimated to take a minimum of 12 weeks to cluster manually. These techniques produce a list of values whose length is equal to the dataset length. Each value in the list represents a cluster number for the same index VOC in the dataset. For example, the same cluster value was given when two points were assigned to be the same VOC. Non-clustered VOCs were given 0 values in the list which indicates a noisy point. However, these techniques were tested using several ε values and Figure 6 illustrates the accuracy of clustering the ground truths compounds using these techniques. The highest accuracy value reached was 96% (± 0.04 at 95% confidence interval) for VOCCluster and 85% for DBSCAN and OPTICS. The average similarity between points in all the clusters was 96.4%.

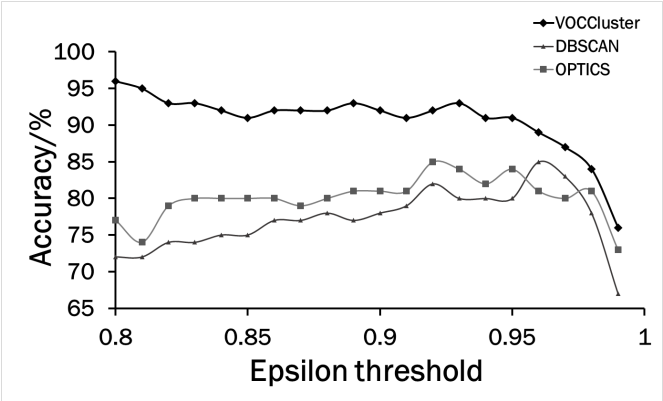


Figure 6: Accuracy of clustering the ground truth VOCs panel of the 74 samples using different clustering techniques with different ε thresholds. VOCCluster was the most accurate technique with an accuracy value of 96% (± 0.04 at 95% confidence interval).

The accuracy of the algorithms varies between compounds depending on the chemical nature of the compound, intensity, extracted m/z profile for that compound etc. In fact, DBSCAN and OPTICS results were influenced by the change of the ε thresholds where many VOCs that were not clustered (FN) when the ε value was increased. Accuracy results for both DBSCAN and OPTICS techniques are reported in the supplementary document Table S5 and S6, respectively. Both techniques demonstrated 95-100% accuracy for highly concentrated compounds such as cyclotrisiloxane, hexamethyl-, toluene and cyclotetrasiloxane octamethyl-, cyclopentasiloxane decamethyl- and cyclohexasiloxane dodecamethyl-. This is because these compounds were well isolated as well as their mass spectra were extracted without missing ions or any other variations, and therefore appeared as dense clusters in the data space that can easily be clustered into groups. Dense points in clusters can significantly contribute to creating distances (drops) between the clusters which makes it easier for them to be distinguished. The accuracy of these distinguishable clusters was not influenced by the different values of ε except at over 0.95.

Alternatively, compounds that appeared in the samples with low or various concentrations were difficult to cluster. This leads to an increase in the distances between a cluster's points in the data space. Some m/z values were missing from some compounds after deconvolution of the samples and hence generating a different mass spectrum profile to other compounds of higher concentration. Non-dense points were spread in the data space (widespread cluster), causing overlapping between clusters or border points might be reachable from more than one cluster. For example, the average extracted ions for benzophenone compound with the low concentration compounds was 3 ions while it was 8 ions for the high concentration. Therefore, it was clustered, using DBSCAN, with a 93% accuracy when ε value = 0.80. The accuracy dropped down to 68% when the ε value was increased to be 0.95, despite the average accuracy was increased. This behaviour indicates that this compound has appeared in the samples in various concentrations.

In contrast to various concentrations compounds, some overlapped clusters such as the clusters of octane and heptane, 2,4-dimethyl-. Heptane, 2,4-dimethyl- showed a sharp rise in the clustering accuracy (from 10%-78%) of DBSCAN when a ε value was increased from 0.80 to 0.96. Acetic acid is another example that was clustered with an accuracy of 38% using ε value of 0.80 due to many FPs (FP=38 for n=74). The accuracy of this compound jumped to 93% when the ε value increased to 0.96. This results in many compounds disappearing (FNs) from the cluster. These compounds would have been border compounds and originally classified into the cluster on the other side of the border. This happens because of the processing order of the points. It is possible that an inaccurate FP will be clustered first and prevent the accurate one from being added. As a result, a global ε value cannot be applied to these clustering techniques.

VOCCluster was more accurate that DBSCAN and OPTICS using the same dataset. The performance accuracies of the clustering of ground truths compounds are presented in Table 1 below. VOCCluster was faster than manual processing (3 hrs *vs* approx. 4 months) and superior to existing computational techniques where studies are usually designed based on visualisation implementations like PCA to determine the number of clusters that can be produced.

Table 1: A summary of VOCCluster performance accuracy based on the "ground truth"(GT) VOC panel.

Level 2 identification (33)	CAS	Manual		VOCCluster				
		GT-TP	GT-TN	TP	FP	FN	TN	Accuracy %
acetone	67-64-1	65	9	65	0	0	9	100
methane-d, trichloro	865-49-6	74	0	74	0	0	0	100
toluene	108-88-3	74	0	74	0	0	0	100
toluene-D8	2037-26-5	74	0	71	0	3	0	95.9
cyclotetrasiloxane, octamethyl-	556-67-2	74	0	70	0	4	0	94.6
cyclopentasiloxane, decamethyl-	541-02-6	74	0	73	0	1	0	98.6
cyclotrisiloxane, hexamethyl-	541-05-9	74	0	74	0	0	0	100
cyclohexasiloxane, dodecamethyl-	540-97-6	74	0	73	0	1	0	100
benzophenone	119-61-9	57	17	53	0	4	17	94.6
[1 1':3' 1"-terphenyl]-2'-ol	2432-11-3	53	21	52	1	1	20	97.3
benzaldehyde	100-52-7	74	0	74	0	0	0	100
benzene	71-43-2	70	4	70	0	0	4	100
nonanal	124-19-6	68	6	66	0	2	6	97
heptanal	111-71-7	49	25	48	4	1	22	93.3
decanal	112-31-2	59	15	57	3	2	12	93.3
ethylbenzene	100-41-4	66	8	58	0	8	8	89.2
à-pinene	80-56-8	65	9	63	1	2	8	95.9
hexanal	66-25-1	48	26	47	5	1	22	92
furfural	98-01-1	38	36	35	0	3	36	95.9
benzofuran	271-89-6	64	10	64	3	0	7	95.9
acetic acid	64-19-7	56	18	56	0	0	18	100
styrene	100-42-5	74	0	74	0	0	0	100
thiophene, 3-methyl-	104-76-7	48	26	44	0	4	26	95
1-hexanol, 2-ethyl-	2213-23-2	59	15	51	6	8	10	81
heptane, 2,4-dimethyl-	111-65-9	65	9	57	1	8	9	88
octane	19549-87-2	31	43	26	2	5	43	91
2,4-dimethyl-1-heptene	616-44-4	63	11	61	1	1	11	97
							Mean VOCCluster Accuracy	96%

Due to the re-clustering mechanism applied in VOCCluster, the results are not sensitive to the order of the points in the dataset. The use of the different ε thresholds did not have an effect on the clustering of the high concentration compounds such as methane-d trichloro- and toluene within a high-density cluster. The same applied for low concentration compounds such as benzene and benzophenone within a widespread cluster. However, when high ε values such as 0.95 were used, the inaccuracy of clustering was evident with these kinds of subsets. This is an impact that is clearly apparent when there were increases in the FNs in each of the clusters.

The accuracy of VOCCluster (Table 1) varies between compounds depending on the chemical nature of the compound, intensity, extracted m/z profile for that compound etc. For example, toluene and siloxanes were the most accurate clusters as they had a good chromatography separation from other compounds, their mass spectra profiles were well defined with all of their m/z values extracted following the deconvolution process. However, this is not the case for other compounds. For example, heptanal was a challenging compound to cluster with n = 4 FN that were observed. These FNs were mainly because of heptanal's mass spectrum profile as the majority of the m/z values were not extracted from the raw data upon deconvolution. This was a result of the overlapping between this compound in these 4 samples with other co-eluting compounds. Toluene-D8 and another feature (m/z 159) were an example of co-eluting compounds, m/z 159 was extracted with the toluene-D8 ions following deconvolution and caused a challenge in the clustering process (supplementary data, Table S7).

Decanal is another challenging compound to detect as it is often misclassified when at low concentrations. Here some m/z values are missing and hence a different mass spectrum profile to the decanal cluster of higher concentration. The average extracted ions for the low concentration compounds was 8 ions while it was 18 ions for the high concentration. However, this needs further clarification by chemists as calibration curves and lower limit of detection for decanal on the GC-MS analytical method are needed to verify the above proposition.

It's also important to note that VOCCluster, with minimum tuning, can be used to cluster features and peaks for other GC-MS metabolomics data matrices such as saliva, skin and urine. Overall, *VOCCluster* provides more accurate results compared with the available approaches in the literature. *VOCCluster* demonstrates a new computational approach to clustering VOCs from breath data, it was able to cluster 15,000 features from 74 clinical GC-MS samples in less than 3 hours on a commodity computer.

CONCLUSION

The VOCCluster algorithm provides untargeted metabolomics feature clustering for breath GC-MS data. VOCCluster was used for a clinical breath data set (n = 74) obtained from cancer patients before and after radiation therapy as part of the TOXI-triage clinical trial. Mass spectra similarities, RI range, cosine similarity, and new clustering principles were optimised and applied to the clinical data set. VOCCluster was evaluated and compared to a manual VOC panel "ground truth", DBSCAN and OPTICS. The accuracy of all three computational approaches varied between compounds depending on their chemical nature of the compound, extracted m/z profiles, intensities, etc. VOCCluster resulted in the most accurate clustering of VOCs (96%, ± 0.04 at 95% confidence interval), while both DBSCAN and OPTICS were ε dependent and demonstrated a maximum accuracy of 85%.

ASSOCIATED CONTENT

Supporting Information

Supplementary information document is provided with this manuscript and it contains the following:

Table S1: List of targeted compounds that were used to calculate the RI range for this clinical breath data set. **Table S2:** Example of a distance matrix for compounds in the clinical radiation dataset. **Table S3:** An example to demonstrate VOCCluster re clustering ability for octane and heptane, 2,4-dimethyl-. **Table S4:** GC-MS Instrumentation parameters. **Table S5:** An illustration of DBSCAN results and the accuracy for each cluster of the ground truth compounds. **Table S6:** An illustration of OPTICS results and the accuracy for each cluster of the ground truth compounds. **Table S7:** An example of VOCCluster and DBSCAN results for toluene-D8 and how for some samples it was co eluting with another feature m/z 159. **Figure S1:** Three-dimensions data output for breath sample after deconvoluted. **Figure S2:** Distance matrix heat-map for 15,307 VOCs illustrating the similarity between VOCs from different samples in the distance matrix. **Figure S3:** Three-dimensional plot of ground truth compounds that were clustered manually. t-Distributed Stochastic Neighbour Embedding (t-SNE) was used to show how compounds with different densities are distributed from the mean of 0. **Figure S4:** Illustration of VOCCluster algorithm 1 (main loop) which examines each object (VOC) in the dataset and initiates a new cluster if the selected VOC is not clustered and has neighbours. **Figure S5:** Illustration of VOCCluster algorithm 2 which was used to extract neighbours for a given VOC. VOC's neighbours are added into the N list and returned into the requested algorithm. **Figure S6:** Illustration of VOCCluster algorithm 3 that was used to grow a cluster with the condition of only having one VOC from a sample in the cluster. **Figure S7:** Illustration of VOCCluster algorithm 4 which is called by algorithm 3. It was used to update properties' values for each clustered VOC. **Figure S8:** Illustration of VOCCluster algorithm 5 which is called by algorithm 3. This function is used to return a VOC that is clustered which is used in the clustering processes. **Figure S9:** Illustration of VOCCluster algorithm 6 which is used to un-label any VOC that was clustered when another VOC from the same sample was found to be closer to the processed cluster and **References**.

AUTHOR INFORMATION

CORRESPONDING AUTHOR

*** Dr Dahlia Salman**

Department of Chemistry, Loughborough University, UK LE11 3TU

Email: D.Salman@lboro.ac.uk

AUTHOR CONTRIBUTIONS

The manuscript was written through contributions of all authors and all authors have given approval to the final version of the manuscript before the submission.

ACKNOWLEDGMENT

The authors of this manuscript would like to acknowledge the TOXI-triage project and the clinical research nurse and radiotherapy staff for their help with obtaining the clinical data sets. TOXI-triage received funding from the European Union's Horizon 2020 Innovation action Programme H2020- EU.3.7. - Secure societies - Protecting freedom and security of Europe and its citizens under grant agreement No 653409. The authors would also like to acknowledge ministry of education in Saudi Arabia for funding the first author to undertake his PhD at the Loughborough University. The authors would also like to acknowledge SpectralWorks limited for providing us with free AnalyzerPro software that was used to deconvolve the clinical breath data used in this study.

REFERENCES

1. Lourenço, C; Turner, C. Breath Analysis in Disease Diagnosis: Methodological Considerations and Applications. *Metabolites*. **2014**,4(2),465–98.

2. Amann, A; Smith, D. Volatile biomarkers: non-invasive diagnosis in physiology and medicine. *Elsevier*; **2013**.

3. Watson, JT; Sparkman, OD. Introduction to mass spectrometry: instrumentation, applications and strategies for data interpretation. *John Wiley & Sons*; **2008**. DOI:10.1002/9780470516898.

4. Rathahao-Paris, E; Alves, S; Junot, C; Tabet, J-C. High resolution mass spectrometry for structural identification of metabolites in metabolomics. *Metabolomics*. **2016**,12(1)),10.

5. Brown, M; Dunn, WB; Ellis, DI; Goodacre, R; Handl, J; Knowles, JD et al. A metabolome pipeline: from concept to data to knowledge. *Metabolomics*. **2005**,1(1),39–51.

6. Kang, S; Paul, Thomas CL. How long may a breath sample be stored for at −80 °C? A study of the stability of volatile organic compounds trapped onto a mixed Tenax:Carbograph trap adsorbent bed from exhaled breath. *J Breath Res*. **2016**,10(2),026011.

7. Guallar-Hoyas, C; Turner, MA; Blackburn,GJ; Wilson, ID; Thomas, CP. A workflow for the metabolomic/metabonomic investigation of exhaled breath using thermal desorption GC–MS. *Bioanalysis*. **2012**,4(18),2227–37.

8. Van Berkel, JJBN; Dallinga, JW; Möller, GM; Godschalk, RWL; Moonen, E; Wouters, EFM; et al. Development of accurate classification method based on the analysis of volatile organic compounds from human exhaled air. *J Chromatogr B*. **2008**,861(1),101–7.

9. Amal, H; Leja, M; Funka, K; Lasina, I; Skapars, R; Sivins, A; Ancans, G; Kikuste, I; Vanags, A; Tolmanis, I et al. Breath testing as potential colorectal cancer screening tool. *International journal of cancer*. **2016,**138(1),229-236.

10. Peng, G; Hakim, M; Broza, YY; Billan, S; Abdah-Bortnyak, R; Kuten, A; Tisch, U; and Haick, H . Detection of lung, breast, colorectal, and prostate cancers from exhaled breath using a single array of nanosensors. *BJC*.**2010** 103(4),542.

11. Shuster, G; Gallimidi, Z; Reiss, AH; Dovgolevsky, E; Billan, S; Abdah-Bortnyak, R; Kuten, A; Engel, A; Shiban, A; Tisch, U et al. Classification of breast cancer precursors through exhaled breath. *Breast cancer research and treatment.* **2011**, 126(3),791-796.

12. Amann, A; Costello, B; Miekisch, W; Schubert, J; Buszewski, B; Pleil, J; Ratclie, N; Risby, T. The human volatiomr: volatile organic compounds (vocs) in exhaled breath, skin emanations, urine, feces and saliva. *Journal of breath research*. **2008**, 875(2),344-348. ,

13. Van den Velde, S; Nevens, F; Steenberghe, D; Quirynen, M et al. GC-MS analysis of breath odor compounds in liver patients. *J Chromatogr B.* **2008**, 875(2),344-348..

14. Ayodele, T. Types of Machine Learning Algorithms. *New Advances in Machine Learning*. **2010**, pages 19-49..

15. Depke, T; Franke, R; and Bronstrup, M. Clustering of ms2 spectra using unsupervised methods to aid the identification of secondary metabolites from pseudomonas aeruginosa. *Journal of Chromatography B*. **2017**,1071,19-28.

16. Zhao, W; Hopke, P; and Prather, K. Comparison of two cluster analysis methods using single particle mass spectra. *Atmospheric Environment*. **2008**, 42(5),881-892..

17. Hauschild, A-C; Frisch, T; Baumbach, JI; and Baumbach, J . Carotta: Revealing Hidden Confounder Markers in Metabolic Breath Proles. *Metabolites*, **2015**,5(2),344..

18. Purkhart, R; Hillmann, A; Graupner, R; and Becher, G. Detection of characteristic clusters in ims-spectrograms of exhaled air polluted with environmental contaminants. *Int J Ion Mobil Spec*.. **2012,**15(2),63-68.

19. Wiwie, C; Baumbach, J; Röttger, R. Comparing the performance of biomedical clustering methods. *Nat Methods*. **2015**,12(11),1033–8.

20. Ren, S; Hinzman, AA; Kang, EL; Szczesniak, RD; and Lu, L. Computational and statistical analysis of metabolomics data. *Metabolomics*. **2015** 11(6),1492-1513..

21. De Souza, DP; Saunders, EC; McConville, MJ; Likic, VA. Progressive peak clustering in GC-MS Metabolomic experiments applied to Leishmania parasites. *Bioinformatics*. **2006**, 22(11),1391–6.

22. Fiehn, O; Kopka, J; N, R. Trethewey and, Willmitzer L. Identification of Uncommon Plant Metabolites Based on Calculation of Elemental Compositions Using Gas Chromatography and Quadrupole Mass Spectrometry. *Anal. Chem*. **2000**,72 (15), 3573–3580.

23. Ester, M; Kriegel, H-P; Sander, J; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. *In Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*.**1996** ,226-231.

24. Kriegel, H-P; Kröger, P; Sander,J; Zimek, A. Density-based clustering. *Wiley Interdiscip Rev Data Min Knowl Discov*. **2011**,1(3),231–40.

25. Rieder, V; Schork, KU; Kerschke, L; Blank-Landeshammer, B; Sickmann,A; Rahnenführer, J. Comparison and Evaluation of Clustering Algorithms for Tandem Mass Spectra. *J Proteome Res*. **2017**,16(11),4035–44.

26. Peng, L; Dong, Z; and Naijun, W. VDBSCAN: Varied Density Based Spatial Clustering of Applications with Noise, *Proc. of IEEE Conference*. **2007**.528-531.27. Depke, T; Franke, R; Brönstrup, M. Clustering of MS2 spectra using unsupervised methods to aid the identification of secondary metabolites from Pseudomonas aeruginosa. *J Chromatogr B*. **2017**, 1071,19–28.

28. Liu, J; Bell, AW; Bergeron, JJ; Yanofsky, CM; Carrillo, B; Beaudrie, CE et al. Methods for peptide identification by spectral comparison. *Proteome Sci*. **2007**,5(1),3.

29. Stein, SE; Scott, DR. Optimization and testing of mass spectral library search algorithms for compound identification. *J Am Soc Mass Spectrom*. **1994**,5(9),859–66.

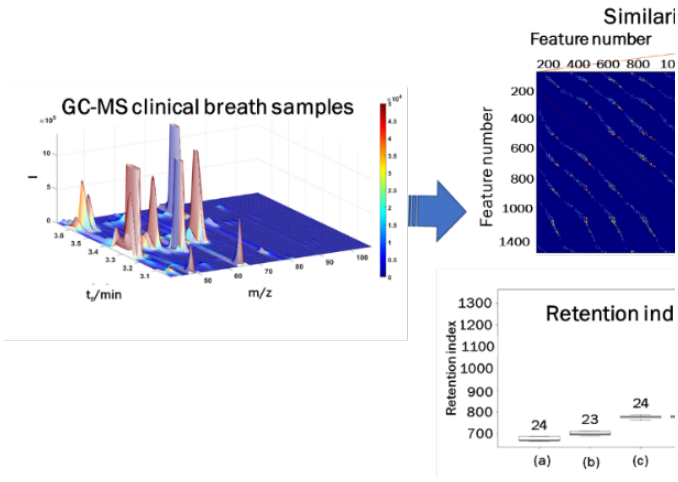
30. Wan, KX; Vidavsky, I; Gross, ML. Comparing similar spectra: from similarity index to spectral contrast angle. *J Am Soc Mass Spectrum*. **2002**,13(1),85–8.

31. TOXI-triage. Tools for detection, traceability, triage and individual monitoring of victims. **2015**. Available from: http://toxi-triage.eu [accessed 2019 Feb 25].

32. Stein, SE. An integrated method for spectrum extraction and identification from gas chromatography/mass spectrometry data. *J Am Soc Mass Spectrom*. **1999**, 10(99),770-781..

33. Salek, RM; Steinbeck, C; Viant, MR; Goodacre, R; Dunn, WB. The role of reporting standards for metabolite annotation and identification in metabolomic studies. *Gigascience*. **2013**,2(1),13.

FOR TOC ONLY


GC-MS clinical breath samples
I
t_R/min
m/z
Similari
Feature number
200 400 600 800 10
200
400
600
800
1000
1400
Feature number
Retention ind
1300
1200
1100
1000
900
800
700
Retention index
24
23
24
(a)
(b)
(c)

Supplementary Materials

VOCCluster: Untargeted Metabolomics Feature Clustering Approach for Clinical Breath Gas Chromatography - Mass Spectrometry Data

Yaser Alkhalifah[1], Iain Phillips[1], Andrea Soltoggio[1], Kareen Darnley[3], William H. Nailon[3], Duncan McLaren[3], Michael Eddleston[2], C. L. Paul Thomas[4], Dahlia Salman[4*]

[1]Department of Computer science, Loughborough University, Loughborough, LE11 3TU, UK.
[2]Pharmacology, Toxicology & Therapeutics Unit, University of Edinburgh, Edinburgh, UK.
[3]Edinburgh Cancer Centre, NHS Lothian, Edinburgh, UK.
[4]Department of Chemistry, Loughborough University, Loughborough, LE11 3TU, UK.

Corresponding author: Dr Dahlia Salman, D.Salman@lboro.ac.uk

Table of content

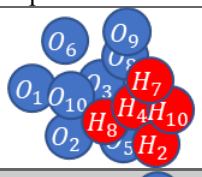
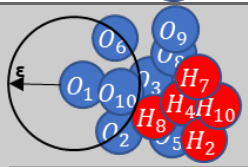
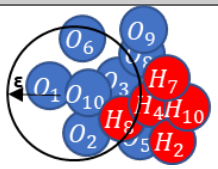
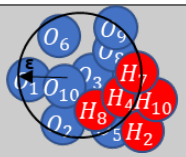
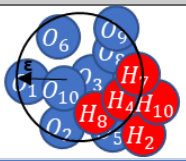
Table S1: List of targeted compounds that were used to calculate the RI range for this clinical breath data set. The following information is needed for each targeted compound: sample number (#), compound number (ID), probability of minimum (RI-) and maximum (RI+) retention index

Level 2 identification (1)	**CAS**	**Sample #**	**ID**	**RI-**	**RI+**
acetone	67-64-1	2	9	660	690
methane-d, trichloro-	865-49-6	6	18	690	720
toluene-D8	2037-26-5	5	43	750	790
toluene	108-88-3	6	40	750	800
cyclotrisiloxane, hexamethyl-	541-05-9	4	55	800	830
cyclotetrasiloxane, octamethyl-	556-67-2	66	105	950	1000
cyclopentasiloxane, decamethyl-	541-02-6	62	171	1110	1160
cyclohexasiloxane, dodecamethyl-	540-97-6	14	225	1280	1340

Table S2: Example of a distance matrix for compounds in the clinical radiation dataset. The VOCs that are out of the RI range for a particular VOC is given 0 similarity, otherwise they will be given 1 similarity, $C_{i,j}$ indicates compound j in sample i in the dataset

	C0,0	C0,1	C0,2	C0,3	C0,4	C0,5	. . .	. . .	C1,0	C1,1	C1,2	C1,3	C1,4	C1,5	. . .
C0,0	1	0	0	0	0	0	. . .	. . .	0.992	0	0	0	0.174	0	. . .
C0,1	0	1	0	0	0	0	. . .	. . .	0	0.998	0	0	0.064	0.996	. . .
C0,2	0	0	1	0	0	0			0	0	0	0	0	0	. . .
C0,3	0	0	0	1	0	0	. . .	. . .	0.79	0	0	0	0.645	0	. . .
C0,4	0	0	0	0	1	0			0	0	0	0.019	0	0	. . .
C0,5	0	0	0	0	0	1	. . .	. . .	0	0	0	0	0	0	. . .
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
C1,0	0.992	0	0	0.790	0	0	. . .	. . .	1	0	0	0	0	0	. . .
C1,1	0	0.998	0	0	0	0	. . .	. . .	0	1	0	0	0	0	. . .
C1,2	0	0	0	0	0	0	. . .	. . .	0	0	1	0	0	0	. . .
C1,3	0	0	0	0	0.019	0	. . .	. . .	0	0	0	1	0	0	. . .
C1,4	0.174	0.064	0	0.645	0	0	. . .	. . .	0	0	0	0	1	0	. . .
C1,5	0	0.996	0	0	0	0	. . .	. . .	0	0	0	0	0	1	. . .
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .

Table S3: An example to demonstrate VOCCluster re clustering ability for octane and heptane, 2,4-dimethyl-

	We have 10 breath samples that we need to cluster their VOCs • 5 out of 10 contain Octane compound only, O • 2 out of 10 contain heptane, 2,4-dimethyl- only, H • 3 out of 10 contain both. Both O and H
 **N** | O_{10} properties: 1 | U | O_1 | U | 1 O_{10} O_6 O_2 O_3 Labelled | core-point | Sampleld *core-distance* | established-distance	O_1 has been selected as an established All VOCs that are neighbours of O_1 listed in N N is sorted based on the high similarity to O_1 Properties of O_1 will be updated when |N|> minPts As O_1 is a stablished point, the core-point will be then itself Undefined (U) will be then given for core-distance and established-distance
 **N** | O_{10} properties: 1 | S(O_1, O_{10}) Cosine similarity | O_1 | S(O_1, O_{10}) | 10 O_3 O_2 H_8 O_6 O_8 H_4	O_{10} will be clusterd with O_1 and its properaties will be updated N will be updated based on the new selected point which is O_{10}
 **N** | O_{10} properties: 1 | S(O_1, O_{10}) | O_1 | S(O_1, O_{10}) | 10 O_8 H_8 O_6 O_9 O_2 H_4 H_7 O_3 properties: 1 | S(O_3, O_{10}) | O_{10} | S(O_1, O_3) | 3	O_3 will be clusterd with O_1 and its properaties will be updated
 H_7 properties: 1 | S(H_4, H_7) | H_4 | S(O_1, H_7) | 7	VOCCluster will keep clustering and cluster will keep growing until: 1- the cluster size reach to the number of samples. Or 2- no more points to be added into the cluster. In this case, points H_8 , H_2 and H_{10} will not be added into the cluster because there are other points from the same samples that have been added. However. points H_4 and H_7 will be added to the O cluster because they were in neighbourhood of other points that were clustered. This what happen with DBACAN and OPTICS as well

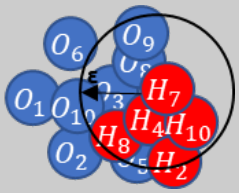
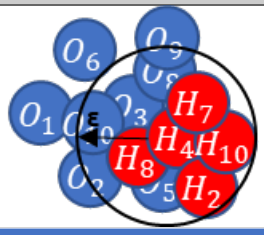
O_{10} properties 1 | S(O_1, O_{10}) | O_1 | S(O_1, O_{10}) | 10 O_3 properties 1 | S(O_3, O_{10}) | O_{10} | S(O_1, O_3) | 3 O_8 properties 1 | S(O_8, O_3) | O_3 | S(O_1, O_8) | 8 H_4 properties 1 | S(H_4, O_8) | O_8 | S(O_1, H_4) | 4	
 N: H_4, O_5, O_3, H_7, O_8, H_{10}, O_9, H_2, H_{10}	Now, new cluster will start by selecting a non-clustered point and set as an established point, which is H_8
 Previous H_4 properties 1 | S(H_4, O_8) | O_8 | S(O_1, H_4) | 4 New H_4 properties 1 | S(H_4,H_8) | H_8 | S(H_8, H_4) | 4	The fist point in N list of H_8 was H_4H_4 was labelled 1 (means it was clusterd with O) Therefore, the stored properties of H_4 will be compared with the new properties. If the similarities in the new properties is greater than the previous one. Then, H_4 will be moved to this cluster. It is clear the similarity between H_8 and H_4 is greater than the previous one, therefore, It will re cluster again and its properties will be updated.

Table S4: GC-MS Instrumentation parameters

Thermal desorption		**Gas chromatography**		**Mass spectrometer**	
Parameters	Setting	Parameters	Setting	Parameters	Setting
Primary desorption time	1 min	He carrier gas flow rate	20 cm^3 min^{-1}	Scan type	Full scan (positive)
Primary desorption flow rate	40 cm^3 min^{-1}	Initial oven temperature	40°C	Mass range	40 – 550 m/z
Primary desorption temperature	300°C	Initial hold time	0 min	Ionisation type	EI
Secondary desorption time	5 min	Oven temperature program	5°Cmin^{-1} to 300°C, hold for 8 min	Scan time	3 scans s^{-1}
Secondary desorption flow rate	50 cm^3 min^{-1}	Total run time	60 min	Transfer line temperature	300°C
Secondary desorption temperature	300°C	Post run temperature	45°C	Quadrupole temperature	150°C
Cold trap flow rate	20 cm^3 min^{-1}	Post run time	0 min	Manifold temperature	230°C
Cold trap temperature	-10°C			Solvent delay time	5 min
Trap heating rate	Max °C min^{-1}				
Trap high temperature	300°C				
Trap hold time	5 min				
Flow path temperature	200°C				
Mode	Spitless				

Table S5: An illustration of DBSCAN results and the accuracy for each cluster of the ground truth compounds using an ε = 80 and 96 with a minPts value of 2. Sensitivity and specificity of clusters are described in Section 2.5.2. GT-TP represents the *TP*s compounds in the ground truths out of the 74 samples where GT-TN is the *TN*s samples

Level 2 identification (1)	CAS	GT-TP	GT-TN	ε = 80					ε = 96				
				TP	*FP*	*FN*	*TN*	Accuracy %	*TP*	*FP*	*FN*	*TN*	Accuracy %
acetone	67-64-1	65	9	45	28	20	0	48	65	7	0	2	0.91
methane-d trichloro-	865-49-6	74	0	74	0	0	0	100	72	0	2	0	0.97
toluene	108-88-3	74	0	74	0	0	0	100	74	0	0	0	1
toluene-D8	2037-26-5	74	0	69	0	5	0	93	49	0	25	0	0.66
cyclotetrasiloxane octamethyl-	556-67-2	74	0	73	1	1	0	97	73	1	1	0	0.97
cyclopentasiloxane decamethyl-	541-02-6	74	0	71	3	3	0	92	72	2	2	0	0.95
cyclotrisiloxane hexamethyl-	541-05-9	74	0	74	0	0	0	100	74	0	0	0	100
cyclohexasiloxane dodecamethyl-	540-97-6	74	0	73	1	1	0	97	70	0	4	0	95
benzophenone	119-61-9	57	17	53	1	4	17	93	34	0	23	17	68
[1 1':3' 1"-terphenyl]-2'-ol	2432-11-3	53	21	52	1	1	21	97	48	1	5	21	92
benzaldehyde	100-52-7	74	0	74	0	0	0	100	72	0	2	0	97
benzene	71-43-2	70	4	70	0	0	4	100	54	0	16	4	78
nonanal	124-19-6	68	6	40	5	28	3	57	59	0	9	6	88
heptanal	111-71-7	49	25	27	5	22	21	64	26	0	23	25	69
decanal	112-31-2	59	15	21	45	38	4	23	36	0	23	15	69
ethylbenzene	100-41-4	66	8	57	16	9	1	70	66	7	0	1	72
α-pinene	80-56-8	65	9	64	2	1	8	96	49	0	16	9	78
hexanal	66-25-1	48	26	31	15	17	15	59	34	1	14	25	80
furfural	98-01-1	38	36	20	7	18	36	69	19	4	19	36	71
benzofuran	271-89-6	64	10	64	3	0	7	96	54	3	10	7	82
acetic acid	64-19-7	56	18	36	38	20	0	38	52	1	4	18	93
styrene	100-42-5	74	0	66	0	8	0	89	59	0	15	0	80
1-hexanol, 2-ethyl-	104-76-7	59	15	13	55	46	2	13	50	0	9	15	88
heptane, 2,4-dimethyl-	2213-23-2	65	9	10	60	55	3	10	57	10	8	7	78
octane	111-65-9	31	43	7	63	24	3	10	19	4	12	39	78
2,4-Dimethyl-1-heptene	19549-87-2	62	12	15	51	47	4	16	55	1	7	11	89
thiophene, 3-methyl-	616-44-4	48	26	48	1	0	25	99	46	0	2	26	97
					Average accuracy			71	Average accuracy				85

Table S6: An illustration of OPTICS results and the accuracy for each cluster of the ground truth compounds using an ε = 81 and 92 (worst and best) with a minPts value of 2. Sensitivity and specificity of clusters were described in Section 3.2. GT-TP represents the *TP*s compounds in the ground truths out of the 74 samples where GT-TN is the *TN*s samples

Level 2 identification (1)	CAS	GT-TP	GT-TN	ε = 81					ε = 92				
				TP	*FP*	*FN*	*TN*	Accuracy %	*TP*	*FP*	*FN*	*TN*	Accuracy %
acetone	67-64-1	65	9	51	23	14	0	58	51	23	14	0	58
methane-d trichloro-	865-49-6	74	0	74	0	0	0	100	74	0	0	0	100
toluene	108-88-3	74	0	74	0	0	0	100	74	0	0	0	100
toluene-D8	2037-26-5	74	0	69	0	5	0	93	73	0	1	0	99
cyclotetrasiloxane octamethyl-	556-67-2	74	0	74	0	0	0	100	74	0	0	0	100
cyclopentasiloxane decamethyl-	541-02-6	74	0	74	0	0	0	100	74	0	0	0	100
cyclotrisiloxane hexamethyl-	541-05-9	74	0	74	0	0	0	100	74	0	0	0	100
cyclohexasiloxane dodecamethyl-	540-97-6	74	0	74	0	0	0	100	74	0	0	0	100
benzophenone	119-61-9	57	17	48	1	9	17	87	48	1	9	17	87
[1 1':3' 1"-terphenyl]-2'-ol	2432-11-3	53	21	52	1	1	21	97	52	1	1	21	97
benzaldehyde	100-52-7	74	0	74	0	0	0	100	74	0	0	0	100
benzene	71-43-2	70	4	69	0	1	4	99	69	0	1	4	99
nonanal	124-19-6	68	6	55	4	13	3	77	64	2	4	5	92
heptanal	111-71-7	49	25	25	0	24	25	68	41	0	8	25	89
decanal	112-31-2	59	15	21	28	38	10	32	55	1	4	14	93
ethylbenzene	100-41-4	66	8	49	25	17	0	54	49	25	17	0	54
α-pinene	80-56-8	65	9	62	1	3	8	95	62	1	3	8	95
hexanal	66-25-1	48	26	26	33	22	15	43	46	17	2	10	75
furfural	98-01-1	38	36	20	3	18	36	73	20	3	18	36	73
benzofuran	271-89-6	64	10	62	3	2	7	93	62	3	2	7	93
acetic acid	64-19-7	56	18	54	18	2	1	73	54	2	2	16	95
styrene	100-42-5	74	0	63	0	11	0	85	71	0	3	0	96
1-hexanol, 2-ethyl-	104-76-7	59	15	29	31	30	2	34	36	26	23	2	44
heptane, 2,4-dimethyl-	2213-23-2	65	9	14	12	51	5	23	20	53	45	1	18
octane	111-65-9	31	43	7	52	24	10	18	11	36	20	19	35
2,4-Dimethyl-1-heptene	19549-87-2	62	12	25	8	37	9	43	61	3	1	10	95
thiophene, 3-methyl-	616-44-4	48	26	48	0	0	26	100	48	0	0	26	100
				Average accuracy				74	Average accuracy				85

Table S7: An example of VOCCluster and DBSCAN results for toluene-D8 and how for some samples it was co eluting with another feature (m/z 159)

Sample #	Compound #	Ion 1	Ion 2	Ion 3	Ion 4	Ion 5	RT	RI	Ions	VOCCluster	DBSCAN
1	45	98	100	42	70	43	5.9	771	18	✓	✓
2	39	98	100	159	43	70	5.9	770	24	✓	
3	68	98	100	48	94	0	5.7	764	4	✓	✓
4	43	98	100	43	70	42	5.7	763	18	✓	✓
5	43	98	100	42	70	54	5.0	780	8	✓	✓
6	38	98	100	42	70	54	5.0	779	9	✓	✓
7	33	98	100	42	70	54	5.7	764	11	✓	✓
8	49	98	100	43	70	71	5.7	764	33	✓	
9	59	98	43	100	70	71	5.7	764	22	✓	
10	37	43	70	71	41	42	6.0	776	23		
11	29	98	100	42	70	54	5.0	778	6	✓	✓
12	43	98	100	42	70	54	5.0	779	12	✓	✓
13	45	43	159	70	71	98	6.1	780	40	✓	
14	45	98	100	42	70	54	5.0	780	9	✓	✓
15	34	43	98	70	100	71	5.7	763	20	✓	
16	50	98	100	42	70	54	4.9	774	20	✓	✓
17	42	98	100	42	70	54	5.0	779	20	✓	✓
18	36	98	100	45	54	66	4.9	775	10	✓	
19	50	98	100	42	70	54	4.8	772	21	✓	✓
20	48	98	100	70	42	54	6.1	778	17	✓	✓
21	50	43	98	70	71	100	6.2	782	44	✓	
22	45	98	100	42	70	54	5.0	780	8	✓	✓
23	50	98	100	43	70	71	5.7	763	15	✓	✓
24	39	98	100	70	42	54	5.7	764	12	✓	✓
25	45	43	98	70	100	71	5.7	764	29	✓	
26	55	98	100	42	70	54	5.7	764	14	✓	✓
27	61	43	70	98	41	71	6.0	775	23	✓	
28	44	98	100	42	70	44	6.0	773	23	✓	✓
29	36	98	100	42	44	70	4.8	772	11	✓	✓
30	48	98	100	70	42	44	5.7	764	15	✓	✓
31	43	98	100	42	70	54	5.0	780	9	✓	✓
32	33	98	100	42	70	54	5.7	764	14	✓	✓
33	40	98	43	100	70	71	6.0	776	18	✓	
34	28	98	100	70	54	66	4.9	775	10	✓	✓
35	48	98	100	42	70	54	4.8	772	17	✓	✓
36	31	98	100	42	70	54	6.1	780	12	✓	✓
37	26	98	100	42	70	54	5.0	779	18	✓	✓
38	65	43	159	70	98	71	6.1	780	42	✓	
39	52	43	159	70	71	98	6.2	781	53	✓	
40	41	98	100	42	70	54	5.0	780	10	✓	✓
41	44	98	100	42	70	54	6.1	780	14	✓	✓

42	37	98	100	42	70	54	6.1	780	13	✓	✓
43	47	98	100	70	42	41	5.7	764	14	✓	✓
44	46	98	100	42	70	43	6.2	783	16	✓	✓
45	39	98	100	42	70	54	5.0	779	20	✓	✓
46	42	98	159	43	100	70	6.0	773	56	✓	
47	40	98	100	42	70	54	5.0	780	8	✓	✓
48	60	98	100	42	48	54	6.3	786	9	✓	✓
49	76	43	98	70	71	100	6.0	774	37	✓	
50	43	98	100	42	70	43	5.7	763	13	✓	✓
51	40	98	100	42	70	54	6.3	787	16	✓	✓
52	30	98	100	42	70	54	5.0	778	7	✓	✓
53	30	43	98	70	71	41	6.1	780	18		
54	32	98	100	42	70	66	5.0	778	6	✓	✓
55	59	98	100	42	70	54	5.0	779	9	✓	✓
56	30	43	70	71	98	41	6.1	778	14		
57	38	98	100	42	70	54	4.9	775	16	✓	✓
58	48	98	100	0	0	0	6.1	779	2	✓	✓
59	20	98	100	42	70	54	6.0	776	9	✓	✓
60	51	98	100	42	64	54	6.3	786	11	✓	✓
61	37	43	159	98	70	71	5.9	773	50	✓	
62	40	98	43	100	70	71	6.1	780	27	✓	
63	36	159	98	43	100	70	5.9	770	56	✓	
64	41	43	98	70	100	71	6.0	776	25	✓	
65	48	98	100	42	70	41	6.0	775	15	✓	✓
66	52	98	43	100	70	71	6.0	774	24	✓	
67	46	43	159	70	71	98	6.1	779	41	✓	
68	32	98	100	42	70	44	5.0	779	21	✓	✓
69	41	98	100	42	70	54	5.0	780	11	✓	✓
70	41	98	100	42	70	54	4.9	775	12	✓	✓
71	53	98	43	100	70	71	5.7	764	18	✓	
72	29	98	100	42	70	54	4.8	772	13	✓	✓
73	39	98	100	42	70	54	5.0	779	9	✓	✓
74	36	98	100	42	70	54	6.0	775	18	✓	✓

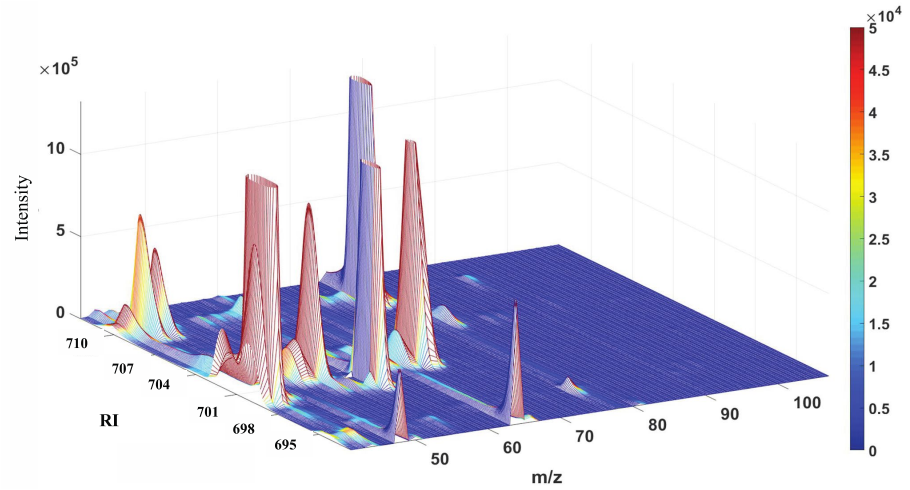


Figure S1: Three-dimensions data output for breath sample after deconvoluted

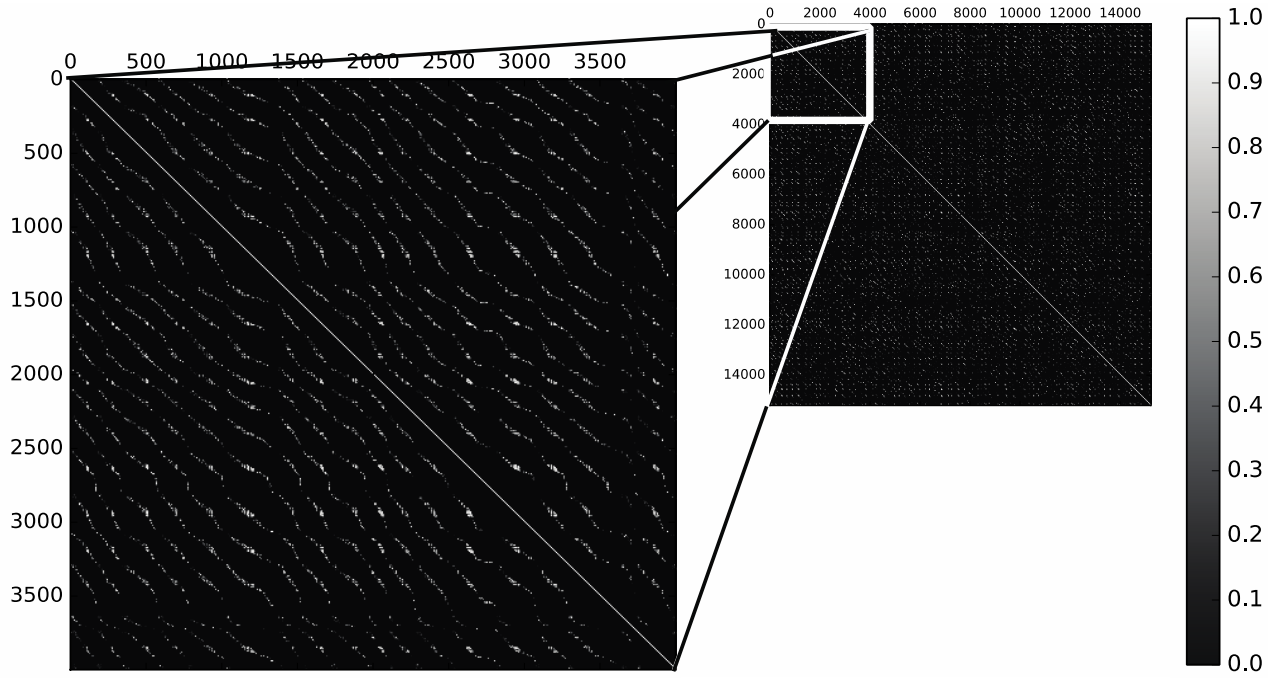


Figure S2: Distance matrix heat-map for 15,307 VOCs illustrating the similarity between VOCs from different samples in the distance matrix. The similarity of 1 (white) means that the two compared VOCs are exactly the same where 0 (black) indicates that they are not similar

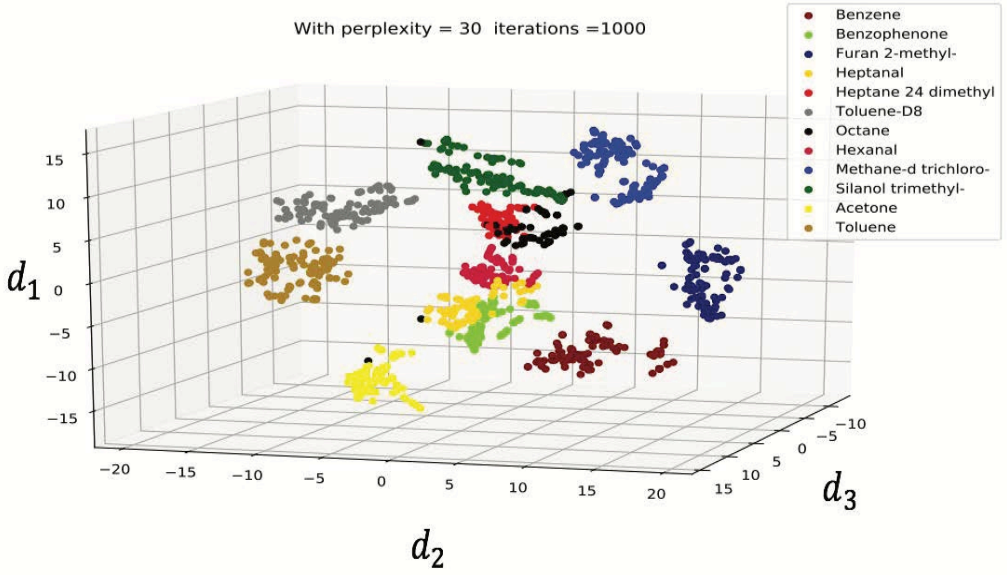


Figure S3: Three-dimensions plot of ground truth compounds that were clustered manually. t- Distributed Stochastic Neighbour Embedding (t-SNE) was used to show how compounds with different densities are distributed from the mean of 0. This plot contains well separated compounds, overlapped compounds and some low densities compounds. These different compounds appeared in different clusters' shapes

Algorithm 1 VOCCluster algorithm

```
# This is the main function that is examining each object in SetOfObjects.
# If an object is not clustered and has neighbours, it initialises a new cluster
# with respect to ϵ and minPts.
# Grow the cluster by giving the neighbours in the cluster the same label.
VOCCluster(SetOfObjects, ϵ, minPts)
    # Cluster counter.
    clusterId = 0
    for each object p in SetOfObjects:
        # Unlabeled previously in the inner loop.
        if p.labelled = undefined:
            neighbours N ← empty list
            # Find p neighbours (N_ϵ(p)), add them into N.
            GetNeighbours(SetOfObjects, N, p, ϵ)
            # Density check.
            if |N| ≥ minPts:
                # Next cluster label.
                clusterId += 1
                # Establish a new cluster; and give its elements same label value.
                NewCuster(SetOfObjects, p, N, clusterId, ϵ, minPts)
```

Figure S4: Illustration of VOCCluster algorithm 1 (main loop) which examines each object (VOC) in the dataset and initiates a new cluster if the selected VOC is not clustered and has neighbours.

Algorithm 2 VOCCluster algorithm - GetNeighbours

```
# This function will extract or update neighbours for a given object.
# All found neighbours will be added into N.
# coreDistance and coreObject are extracted for each object in N.
# N is updated if less distance value to a new object has been discovered.
GetNeighbours(SetOfObjects, N, currentObject, ε)
    # Initial point check.
    if N is empty list:
        # Scan all objects in the SetOfObjects.
        # Add density-reachable objects to the currentObject into N list.
        for each object p in SetOfObjects:
            # Compute objects' mass spectra similarity and check threshold.
            if currentObject.dist(p) ≤ ε:
                # Copy object p into a new object q, and calculate q's properties.
                q ← p
                q.coreDistance ← currentObject.dist(p)
                q.coreObject ← currentObject
                # Add q into neighbour's list.
                N ← N ∪ q
    else:
        # The currentObject is density-reachable from previous coreObject.
        # Scan the SetOfObjects to add, update objects.
        # Add density-reachable objects to the currentObject not in N.
        # Update density-reachable object that is in N,
        # which is more close to the currentObject than the previous coreObject.

        for each object p in SetOfObjects:
            if currentObject.dist(p) ≤ ε:
                if p ∉ N:
                    # Object p is not in N.
                    # Copy p object into a new object q.
                    q ← p
                    q.coreDistance ← currentObject.dist(p)
                    q.coreObject ← currentObject
                    # Add q into neighbour's list.
                    N ← N ∪ q
                else:
                    # Object p is in N.
                    for each object q in N:
                        # Find p in N → q.
                        # Check if p is close to the currentObject more than,
                        # q to the coreObject (q.coreObject), if so, update q.
                        if p = q and q.coreDistance > currentObject.dist(p):
                            q.coreDistance ← currentObject.dist(p)
                            q.coreObject ← currentObject
                            break
    # Reorder the neighbours based on the coreDistance value.
    Sort N by coreDistance.
```

Figure S5: Illustration of VOCCluster algorithm 2 which was used to extract neighbours for a given VOC. VOC's neighbours are added into the N list and returned into the requested algorithm.

Algorithm 3 Algorithm VOCCluster - NewCuster

```
# This function to grow cluster by given discovered objects the same label.
# A cluster only contains one object from the same sample.
NewCuster(SetOfObjects, establishedObject, N, clusterId, ϵ, minPts)
    Samples S ← empty list
    establishedObject.labelled ← clusterId
    S ← establishedObject.sampleId
    while N is not empty:
      q ← N.pop(0)
      if q.labelled = undefined & q.sampleId ∉ S:
        ClusterObject(SetOfObjects, q, establishedObject, clusterId, N,
          ϵ , minPts)
        S ← S ∪ q.sampleId
      elif q.labelled = undefined & q.sampleId ∈ S:
        ObjInCluster ← GetObject(SetOfObjects, q, clusterId)
        if q.coreDistance < ObjInCluster.coreDistance &
          establishedObject.dist(q) ≤ ObjInCluster.establishedDistance:
          Release(SetOfObjects, ObjInCluster)
          ClusterObject(SetOfObjects, q, establishedObject, clusterId, N,
            ϵ , minPts)
      else:
        ClusteredObj ← GetObject(SetOfObjects, q, q.labelled)
        if q.coreDistance < ClusteredObj.coreDistance &
          establishedObject.dist(q) ≤ ClusteredObj.establishedDistance:
          if ClusteredObj.labelled = clusterId:
            ClusteredObj.coreDistance ← q.coreDistance
            ClusteredObj.coreObject ← q.coreObject
          elif q.sampleId ∈ S:
            Release(SetOfObjects, ClusteredObj)
            ObjInCluster ← GetObject(SetOfObjects, q, clusterId)
            if q.coreDistance < ObjInCluster.coreDistance
              & establishedObject.dist(q) ≤
                ObjInCluster.establishedDistance:
              Release(SetOfObjects, ObjInCluster)
              ClusterObject(SetOfObjects, q, establishedObject, clusterId,
                N, ϵ , minPts)
          else:
            Release(SetOfObjects, ClusteredObj)
            ClusterObject(SetOfObjects, q, establishedObject, clusterId, N,

              ϵ , minPts)
            S ← S ∪ q.sampleId
```

Figure S6: Illustration of VOCCluster algorithm 3 that was used to grow a cluster with the condition of only having one VOC from a sample in the cluster.

Algorithm 4 Algorithm VOCCluster - Cluster Object

```
ClusterObject(SetOfObjects, q, establishedObject, clusterId, N, ϵ , minPts)
    for each object p in SetOfObjects:
        if p = q:
            p.labelled ← clusterId
            p.coreDistance ← q.coreDistance
            p.coreObject ← q.coreObject
            p.establishedDistance ← establishedObject.dist(q)
            neighbours qN ← empty list
            GetNeighbours(SetOfObjects, qN, q, establishedObject, ϵ)
            if |qN| ≥ minPts:
                GetNeighbours(SetOfObjects, N, q, establishedObject, ϵ)
            break
```

Figure S7: Illustration of VOCCluster algorithm 4 which is called by algorithm 3. It was used to update properties' values for each clustered VOC.

Algorithm 5 Algorithm VOCCluster - Get Object in cluster

```
GetObject(SetOfObjects, q, clusterId)
    for each object p in SetOfObjects:
        if p.labelled = clusterId & p.sampleId = q.sampleId:
            return p
```

Figure S8: Illustration of VOCCluster algorithm 5 which is called by algorithm 3. This function is used to return a VOC that is clustered which is used in the clustering processes.

Algorithm 6 Algorithm VOCCluster - Release

```
Release(SetOfObjects, ObjInCluster)
    ReleaseObjects R ← empty list
    R ← ObjInCluster
    while R is not empty:
        q ← N.pop(0)
        for each object p in SetOfObjects:
            if p = q:
                p.labelled = undefined
                p.coreDistance = undefined
                p.coreObject = undefined
                p.establishedDistance = undefined
            elif p.coreObject = q:
                R ← R ∪ p
```

Figure S9: Illustration of VOCCluster algorithm 6 which is used to un-label any VOC that was clustered when another VOC from the same sample was found to be closer to the processed cluster.

References

1. Salek RM, Steinbeck C, Viant MR, Goodacre R, Dunn WB. The role of reporting standards for metabolite annotation and identification in metabolomic studies. Gigascience. 2013;2(1):13.